



Europäisches Patentamt
European Patent Office
Office européen des brevets



11) Publication number: 0 590 995 A2

12)

EUROPEAN PATENT APPLICATION

21) Application number: 93307806.5

51) Int. Cl.⁵: G06F 15/72

22) Date of filing: 30.09.93

30) Priority: 02.10.92 GB 9220767

43) Date of publication of application:
06.04.94 Bulletin 94/14

84) Designated Contracting States:
DE FR GB IT

71) Applicant: CANON KABUSHIKI KAISHA
30-2, 3-chome, Shimomaruko, Ohta-ku
Tokyo (JP)

72) Inventor: Billyard, Adam M., c/o Canon Res.
Ctr Europe Ltd.
Units 17/20, Frederick Sanger Rd, Surrey Res.
Park
Guildford, Surrey GU2 5YD (GB)
Inventor: Huddy, Richard S., c/o Canon Res.
Ctr Europe Ltd.
Units 17/20, Frederick Sanger Rd, Surrey Res.
Park
Guildford, Surrey GU2 5YD (GB)

74) Representative: Beresford, Keith Denis Lewis
et al
BERESFORD & Co. 2-5 Warwick Court High
Holborn
London WC1R 5DJ (GB)

54) Processing image data.

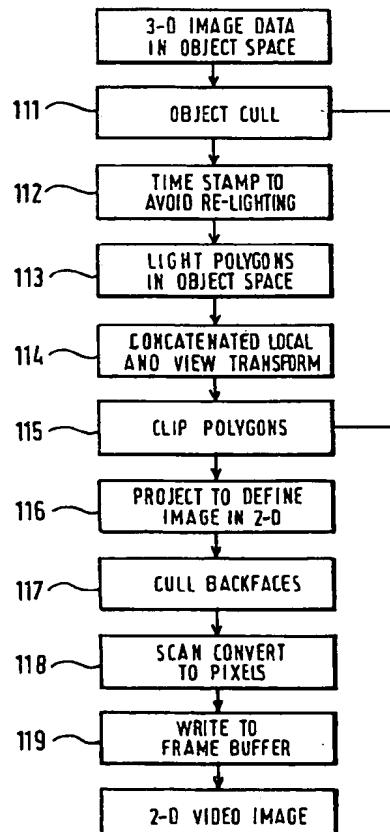
57) 2-dimensional image data is produced from multidimensional image data. The 2-dimensional image data represents a view of the 3-dimensional model from a particular view point.

In the 3-dimensional model, data is stored relating to the position of vertices, therefore, for a particular view point it is not obvious as to whether a polygon is forward facing or back facing.

Vectors defining the polygon edges are processed to determine whether the polygon is forward facing or back facing.

Thus, by processing the vectors defining the polygon it is not necessary to transform normal vectors and a computational saving is made. Further computational savings may be made if vector processing to determine the orientation of the polygon is also required for other processes, such as scan conversion to coloured pixels.

FIG. 11



EP 0 590 995 A2

The present invention relates to apparatus and methods for processing image data.

In particular, the present invention relates to apparatus and methods for processing image data for use in interactive 3-dimensional graphics environments. The present invention also relates to apparatus and methods for generating 2-dimensional representations of 3-dimensional objects, and to image signals and recordings resulting therefrom.

Systems are known which are capable of synthesizing 2-dimensional images in response to data defining elements within a 3-dimensional space. The final 2-dimensional result may consist of a very large number of coloured picture elements (pixels) which may be viewed on a monitor or printed onto an image carrying medium.

In interactive systems arranged to generate data representing a 3-dimensional space, objects appear to move within the 3-dimensional space in response to input commands. Thus, in such systems a machine is required to render a 2-dimensional image from data representing a 3-dimensional space and, in addition, the machine is also required to perform this operation repeatedly as the position and/or orientation of objects, light sources and the view point change in response to input commands. Typically, in an interactive environment, the machine is required to produce output images at a rate of between five to fifteen per second. In more highly powered environments, output images are produced at video rate (60 frames per second) and the machine is said to operate in real time.

In known machines, a detailed example of which will be described later, a significant amount of hardware is required if acceptable images are to be produced at an interactive rate. Thus, computational demands have placed significant constraints on 3-dimensional interactivity, thereby limiting the extent to which interactive 3-dimensional graphics may be employed.

There is a general desire to provide apparatus and methods for processing image data, which require reduced hardware and computational overheads. In particular there is a desire to produce acceptable images at an interactive rate but with lower computational demands.

In known 3-dimensional graphics processing systems, 2-dimensional data is generated by projecting the 3-dimensional image. For each projected polygon, a test is made at some stage as to whether the polygon is forward facing or back facing, whereby it can be eliminated from further processing. Known techniques are described for example by Foley, van Dam, Feiner and Hughes in "Computer Graphics: Principles and Practice" (Second Edition), published by Addison Wesley, ISBN 0-201-12110-7, see particularly pages 663 and 664. Another known system is described in EP-A-0 531 157, not published at the priority date of the present application.

In conventional systems, unit normal vectors for the polygon surfaces have been transformed into viewing space in earlier processes, and can be compared with a viewing direction to determine whether a polygon is forward facing or back facing. On the other hand, transforming and re-normalising unit normal vectors involves a significant computational overhead.

It is an object of the present invention to provide an improved system. In particular, it is an object of the present invention to provide an improved system in which it is unnecessary to transform polygon normal vectors.

According to an aspect of the present invention, there is provided an apparatus including means for processing projected vectors defining polygon edges in screen space to determine whether a polygon is forward facing or back facing. Thus, in the present invention it is unnecessary to transform and re-normalise polygon normals in order to determine whether a polygon is forward facing or back facing. Moreover, in some circumstances, the invention provides accurate results without the need for specific perspective correction measures which are often required in known systems.

The invention further provides a method of generating 2-dimensional image data from data defining objects in a multi-dimensional image space, the method including the step of projecting 2-dimensional polygons into a 2-dimensional screen space, each projected polygon being forward facing or back facing, the method being characterised by processing projected vectors defining polygon edges in screen space to determine whether a polygon is forward facing or back facing.

Said vectors may share a common vertex, and the embodiment may make use of the fact that in the definition of each polygon, the vertices and hence edge vectors of that polygon are always stored in order, in accordance with a predetermined sense of rotation. For example, when facing the front surface of a polygon, the vertices may be listed in an anti-clockwise order.

In the processing of said vectors, the gradients of the vectors may be compared, or alternatively a partial cross product of said vectors may be calculated. The calculated gradients may be useful for subsequent processing, in particular for the scan conversion of polygons which are found to be front facing.

Other features and advantages of the invention will be apparent from a consideration of the embodiments described below. The invention yet further provides image signals and recordings in whatever form that have been generated in accordance with the principles of the invention as set forth above.

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 illustrates an environment for implementing embodiments of the present invention; Figure 2 illustrates a conventional approach to interactive 3-dimensional graphics; Figure 3 represents an object fabricated from polygons in its own object space; Figure 4 represents an object repeatedly transformed into modelling space, including a light source and a viewing position; Figure 5 illustrates techniques for calculating lighting characteristics from light source data, viewing data and surface parameters; Figure 6 shows a viewing space for an isometric projection; Figure 7 shows a frustum providing the first stage of a viewing transformation for perspective views; Figure 8 shows a viewing space derived from the frustum of Figure 7 for generating perspective views; Figure 9 illustrates a projection of polygons into 2-dimensions; Figure 10 illustrates a projected polygon for the purpose of scan conversion; Figure 11 illustrates a process facilitating interactive 3-dimensional graphics production and embodying the present invention; Figure 12 illustrates a bounding box defining extents of an object defined in its own object space; Figure 13 illustrates the transformation of the bounding box shown in Figure 12 into viewing space; Figure 14 illustrates the calculation of lighting characteristics in the process of Figure 11; Figure 15 illustrates algorithms used for calculating diffuse and specular reflections in the process of Figure 11; Figure 16 illustrates a colour look up table for converting eight bit colour values into full colour values for display on a visual display unit, including a plurality of colour ramps; Figure 17 illustrates an alternative colour ramp with gamma correction for luminance values; Figure 18 illustrates an alternatively shaped colour ramp with non-linear saturation values; Figure 19 illustrates schematically an alternative colour ramp combining non-linear luminance and saturation values; Figure 20 illustrates three primary colour component ramps suitable for implementing the ramp of Figure 19; and Figure 21 illustrates the calculation of pixel colour values using an alternative colour look-up table.

A CONVENTIONAL INTERACTIVE GRAPHICS SYSTEM

An environment for synthesizing 3-dimensional

data, viewing 2-dimensional data derived from said 3-dimensional data and interactively modifying said 3-dimensional data is shown in Figure 1. A processor 15 is arranged to write data to and read data from a memory device 16. Data stored in the memory device 16 may define 3-dimensional image data, 2-dimensional image data, instructions to the processor 15 and other types of data which may or may not be relevant to interactive 3-dimensional graphics.

5 In addition, the processor 15 receives input data from input devices 17, said devices consisting of a manually operable keyboard and a position sensitive device, such as a mouse, a tracker-ball or a digitising tablet with a stylus etc.

10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10100 10105 10110 10115 10120 10125 10130 10135 10140 10145 10150 10155 10160 10165 10170 10175 10180 10185 10190 10195 10200 10205 10210 10215 10220 10225 10230 10235 10240 10245 10250 10255 10260 10265 10270 10275 10280 10285 10290 10295 10300 10305 10310 10315 10320 10325 10330 10335 10340 10345 10350 10355 10360 10365 10370 10375 10380 10385 10390 10395 10400 10405 10410 10415 10420 10425 10430 10435 10440 10445 10450 10455 10460 10465 10470 10475 10480 10485 10490 10495 10500 10505 10510 10515 10520 10525 10530 10535 10540 10545 10550 10555 10560 10565 10570 10575 10580 10585 10590 10595 10600 10605 10610 10615 1

and Z axes is positioned at the back bottom left and the vertices of polygons are given x, y and z Cartesian coordinates with reference to this origin.

In Figure 3, a simple object is shown, made up of five polygons. Data is stored in memory defining the position of each vertex of each polygon. For example, polygon 32 has four vertices 34, 35, 36 and 37, the coordinates of which are listed in a table. By convention, the vertices of a polygon are defined in an order obtained by traversing around the polygon in an anti-clockwise direction.

In addition to this positional information, additional information relating to the polygon is also stored. This includes a definition of a unit normal vector 38, a vector extending from the centre of the polygon in a direction perpendicular, i.e. normal to the plane of the polygon, away from its front face, and extending a length equal to unity within the coordinate reference frame. Furthermore, data is also stored relating to the nature of the surface of the polygon and in particular parameters are stored defining how the polygon will react with modelled light directed thereon.

Within a modelling space, in addition to objects, light sources are also provided and a viewing position is selected. Light sources model light falling upon the polygons, which in turn have lighting characteristics, that is to say, the polygons have colour and brightness responsive to the amount of light falling upon them and the nature of their surface.

Thus, within the table defining the polygons of the object shown in Figure 3, data is stored relating to a coefficient of ambient light K(a), a coefficient of diffuse light K(d) and a coefficient of specular light K(s).

The coefficient of ambient light effectively defines how bright a polygon is when no local light sources are provided. The coefficient of diffuse light is a parameter which defines the polygon's characteristic relating to the diffusion of light in response to a local light source. Similarly, the K(s) defines characteristics relating to specular highlights. Thus, the presence of specular highlights effectively represents how shiny a polygon is and improves realism.

Returning to the system detailed in Figure 2, the 3-dimensional image data in object space, as shown in Figure 3, undergoes a local transformation which transforms objects into modelling space, at step 21. At step 22, the polygon surfaces are lit by processing data relating to the position of light sources in modelling space and the lighting parameters of each polygon. Thereafter, at step 23, the modelling space is transformed into a viewing space, the geometry of which differs depending upon whether an isometric view or a perspective view is required.

The view transformation will identify a particular field of view, which will usually cover less than the whole modelling space. Therefore, at step 24 a clipping process is performed to remove polygons which

fall outside the field of view.

Up until this stage, data processed by the processor and read from the memory defines 3-dimensional coordinate locations. At step 25 the projected view is analyzed to define an image in 2-dimensions, thus, in isometric views, the z dimension is rejected and in perspective views the x and y dimensions are scaled, so as to take account of the rejected z dimension.

5 After projecting the image into 2-dimensions at step 25, it is necessary to identify the difference between front faces of polygons and back faces of polygons so that the back faces may be removed, given that they cannot actually be seen because they are obscured by a front face. Thus, at this stage, vertices are defined in 2-dimensions identifying the front faces of visible polygons.

10 At step 27, the 2-dimensional data relating to polygon position is scan converted to produce coloured pixels, taking into account data defining the colour of each polygon, as previously calculated.

15 At step 28, the pixels are written to the frame buffer on a polygon-by-polygon basis, thereby building up a complete 2-dimensional image. It should be noted that images must be written to the frame buffer at a rate of between ten to fifteen frames per second in order that a viewer will feel that the system is responding in an interactive way to input commands.

20 A representation of modelling space is shown in Figure 4. World coordinate axes X, Y and Z have a similar orientation to the axes provided in object space, as shown in Figure 3, but may be scaled differently. For example, it may be appropriate, under default conditions, to define objects as being relatively large in object space and relatively smaller in modelling space, so that a plurality of objects may be assembled in modelling space.

25 In the example of modelling space illustrated in Figure 4, the object illustrated in its own object space of Figure 3, has been instantiated into the modelling space three times. Furthermore, some of the objects have been rotated, so that some normals 38 extend from the centre of polygon 32 are pointing in the positive z direction, whereas, in object space, they were pointing in the positive X direction.

30 Transformations from an object space, such as that illustrated in Figure 3, to a modelling space, such as that illustrated in Figure 4, are effected by matrix manipulation. Thus, the coordinate positions of, say, points 34, 35, 36 and 37 are multiplied by a transform matrix which replaces coordinates within the object space of Figure 3 with coordinates of modelling space as in Figure 4. Three instantiations of the object are placed into modelling space by using three different matrix transforms. Furthermore, additional objects may be placed into modelling space by effecting transformations upon other objects defined in their own object space.

Light sources may be introduced into the modelling space which, in addition to having position, also have orientation. Furthermore, parallel light may be introduced into the modelling space, representing the effect of light emanating from an infinitely displaced source, similar to light produced by the sun. The effect of each light source is considered in turn and lighting effects for each source are added together on a polygon-by-polygon basis.

In addition to the position of light sources, a viewing position 42 is also selected, illustrated by an eye symbol. The viewing position may, alternatively, be represented by a set of axes x, y, z 43, given that a transformation from modelling space as shown in Figure 4, consists of transforming objects into a new space, which has a different set of axes. In particular, whereas, in modelling space, the origin is at the back bottom left, in viewing space the origin is at the front bottom left or front top left. Thus, the z direction represents distances measured away from the viewing position.

As shown in Figure 2, lighting calculations are made at step 22 before the viewing transformation is effected at step 23.

Lighting calculations for polygon 32b will be considered, with reference to Figure 5. The lighting calculation produces an illumination factor I which, in a colour system, consists of colour components RGB.

The value I(RGB) representing the total illumination factor for the polygon, consists of an illumination factor I(a) from ambient lighting, a component I(d) from diffuse lighting from a light source and a component I(s) for specular lighting from said light source. I(a) is derived from the background ambient lighting B(a) and is obtained by multiplying this value by the ambient coefficient K(a), as shown in Figure 5.

I(d) and I(s) are dependent upon the position and intensity of the light source 41 and, in accordance with the Phong model, I(s) is dependent upon the position 42 of the observer. The Phong model is convenient in systems of this type, although other models are known.

In the model, the diffuse component I(d) is derived from the cosine of the angle theta between a vector pointing towards the light source 41 and a vector normal to the plane. Its value is calculated by the scalar or "dot" product of the unit normal vector N and a unit vector L pointing in the direction of the light source 41. Thus, the component I(d) is obtained by multiplying a value identifying the intensity of the light emitted by the light source 41 I(1) by the dot product of vectors N and L by the coefficient K(d).

In accordance with the Phong model, the intensity of specular highlights is dependent upon the position of the viewer 42. A unit vector H is calculated which is half way between the unit vector L and a unit vector E, directed towards the position of the viewer 42. The dot product is then formed between the unit

normal vector N and the unit half way vector H. The model is arranged such that the specular highlight is large when vectors E and L form similar angles to the plane 36, 37, thus resulting in vectors N and H being coincident. Moving away from this position results in the intensity of the specular highlight decreasing rapidly. To represent this in the model, the dot product of the H and N is raised to a power, typically lying within the range 15 to 20 or higher. This value is then multiplied by the coefficient K(s) and the intensity value I(I).

As previously stated, the total intensity value for the polygon in RGB space is calculated by summing I(a) with I(d) and I(s).

Raising the dot product of H and L to a power is a computationally expensive process, therefore the inclusion of means for calculating specular highlights is a significant overhead and in many systems is not included. However, specular highlights add significantly to the realism of the object and allow distinctions to be made between shiny and matt surfaces.

The lighting calculations illustrated in Figure 5 are repeated for each polygon at step 22 of Figure 2. As previously stated, at step 23 a view transformation is made, taking account of the view position 42. It should be noted that, in response to interactive input, the view position 42 may be interactively modified, therefore in the same way in which local transformations and the lighting of surfaces are performed at the interactive rate, view transformation is also performed at the interactive rate.

Most view positions result in only part of the modelling space being viewed, therefore, essentially, a viewing transformation consists of defining the viewable space and shifting the axes.

A representation of a transformation of the modelling space, shown in Figure 4, into isometric viewing space is shown in Figure 6. As can be seen, each vertex is measured from the bottom left corner of a front viewing plane 61 and a viewer is positioned normal to the viewing plane in a minus z direction.

A perspective view transformation may be considered in two stages. As shown in Figure 7, the viewable portion of the modelling space is considered to be a frustum, bounded by a front viewing plane 71, a back viewing plane 72 and sloping side planes 73, 74, 75 and 76. In order for the objects positioned within the frustum to be maintained in their original shape, the space itself is effectively non-Cartesian, therefore further transformation is required, such that the viewing space itself can be represented in Cartesian coordinates. This results in a further transformation of the type illustrated in Figure 8. Herein, the frustum itself has been transformed into a parallelepiped, by transformation under which the extent of negative enlargement increases along the z axis. This results in negative enlargements being made to the objects within the space, such that edges that were par-

allel taper towards a vanishing point as they move in the positive z direction. Thus, data generated under an isometric transformation, of the type illustrated in Figure 6, may now be processed in a similar manner to data generated under a perspective transformation of the type shown in Figure 8.

Referring back to Figure 2, after the 3-dimensional view transformation has been made, it is necessary to perform a clipping operation at step 24 such that polygons which are not included within the viewable space are rejected and do not undergo further processing.

Referring to Figure 8, it can be seen that polygons will lie outside the viewable space. These polygons are identified by comparing their x, y and z coordinates with the coordinates of the viewable space, resulting in a list of polygons which will require further processing.

Referring to Figure 2, the first stage of this further processing is to obtain a 2-dimensional projection of the 3-dimensional data by considering the image that would be seen at the front bounding plane 61/81.

Considering the central object shown in Figure 8, four polygons are visible from the viewing position as shown in Figure 9.

Vertices 94, 95 and 96 represent the vertices of the back face of the object which, as shown in Figure 8, has been reduced in size due to the perspective view transformation.

In addition, to transforming the vertices under the local and view transformations, the unit normal vectors are also transformed. Thus, the front polygon has a unit normal vector extending in the negative z direction of the view coordinates, while the back face has a normal vector extending in the positive z direction. Thus, at step 26 back face culling is performed by rejecting all polygons which have a unit normal vector with a positive z component.

It will be appreciated, therefore, that the unit normal vectors are used at both step 22 when calculating lighting and at step 26 when back face culling. Consequently, the unit vectors are transformed under the local transformation at step 21 and under the view transformation at step 23. Furthermore, these transformations will often produce normal vectors which are no longer of unit length. Consequently, it is necessary to re-normalise these vectors so that, in accordance with the coordinate system being used, they are re-established as normal vectors of unit length. Again, this is computationally expensive, given that square roots must be calculated for each polygon in order to calculate the x, y and z components of the transformed vector, having overall unit length.

Referring to Figure 2, after the back faces have been culled, step 27 consists of scan converting the polygons to produce pixel values.

An irregular polygon is shown in Figure 10 which

has been projected into a 2-dimensional x-y plane. The polygon has not been culled because its front face is showing, therefore descending the list of data coordinates results in anticlockwise traversing around the polygon.

The x-y coordinates in the 2-dimensional projected plane define the positions of pixels within the frame buffer. Thus, the x and y coordinate positions typically lie over a range of 0 to 1,000 in both the x and y dimensions, from an origin at the top left corner.

Scan conversion is initiated by identifying which vertex of the polygon has the smallest y value, in this case, vertex 101. Thus, pixels within the polygon are filled-in starting from this point by calculating the gradients to vertices 102 and 103, which are located at positions (1, 8) and (12, 6) respectively. Thus, as known, it is possible to calculate which pixels within the polygon are to be painted with the polygon colour. This data is then written to the frame buffer at step 28.

Before data is written to the frame buffer, it is conventionally necessary to perform data truncation, given that the number of bits available within the frame buffer is limited, typically to eight bits per pixel location. Thus, a degree of colour aliasing must be tolerated, the extent of which being dependent upon the type of output being used.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

The processes previously described are successfully employed in interactive 3-dimensional graphics systems, in which 3-dimensional image data is modified in response to input commands and output 2-dimensional image data is generated at a rate sufficient to provide feedback to the operator, thereby completing the interactive environment.

The overriding problem with interactive 3-dimensional graphic systems is that of handling very large amounts of data. Each of the processes previously described can be implemented without difficulty and the computational requirements for performing any of the operations once is very modest. However, problems occur when the processing must be repeated many times.

Such a situation lends itself to processing within a hardware pipeline, in which a modest amount of processing is performed at each stage in the pipeline, whereafter the data is clocked to the next stage, resulting in a significant amount of parallel processing. Thus, the system shown in Figure 2 may be implemented as a hardware pipeline in which each process 21 to 28 is performed by its own dedicated hardware elements.

The amount of data being processed tends to increase dramatically as it moves along the stages, particularly given that data which represents the positions of vertices defining a surface must be process-

ed to form data representing the colour of all pixels within that surface. Thus, very large amounts of data are generated during step 28, during which pixel data is written to the frame buffer.

Hardware savings can be made by reducing, wherever possible, the level of data movement and data processing. It should be noted that, when a modest amount of processing is being performed on a large amount of data, even the modest process of moving data from one location to another or comparing a stored value with another value results in a significant overhead, if all of the data under consideration must be moved in this way.

By implementing economies, the object is to reduce the number of physical processors required to implement the necessary process steps. Each saving is significant and if a sufficient number of savings can be made, the whole process can be carried out on a common serial processor.

The operational stages performed by apparatus for processing image data embodying the present invention is shown in Figure 11. Comparing the system of Figure 11 to the conventional system shown in Figure 2, 3-dimensional image data defining object space is read at the start and 2-dimensional video image data is generated at the end, in both systems. However, significant differences exist in the way in which the data is processed, with an overall emphasis being placed on making economies in data transfer and data processing.

At step 111 object culling is performed by determining, after local transformation and viewing transformation, the extent to which an object can actually be seen. Extents are defined for the object in each dimension, that is to say, maximum and minimum values are determined for the object in each of the X, Y and Z dimensions. This process effectively defines a bounding volume in the form of a cuboidal bounding box of planes of constant X, Y or Z extent. Of course, other shapes of bounding volume may conveniently be employed. In particular, a bounding sphere can be defined very economically by reference only to the coordinates (X, Y, Z) of its centre and the magnitude of its radius.

It is not necessary to perform lighting upon the bounding box, therefore the local transformation and viewing transformation are performed as a common operation by concatenating the local and viewing matrices, thereby placing the object's bounding box within viewing space. Thereafter, the position of a transformed bounding box is considered, to determine whether it is totally within the viewable space, totally outside the viewable space or intersecting the viewable space.

If an object is totally outside the viewable space no further processing is performed on the polygons defining such an object, thereby making a significant saving in processing time. For objects totally within

the viewing space, processing continues on these polygons and no saving is made, however, the processing overhead for transforming the bounding box is quite modest, at least in comparison with transforming an object comprising many polygons. For objects which are partially within the viewable space and partially outside the viewable space, data defining the planes of intersection are stored for reconsideration at step 115, involving the clipping of polygons.

5 In the process shown in Figure 2, lighting calculations are performed for each iteration (each new image in the interactive sequence), irrespective of whether they are actually necessary. However, in the present embodiment a clock is present which effectively gives a unique time stamp to each iteration. This stamp is then applied each time lighting is moved and each time surfaces are moved. Thus, if on a present iteration, it can be seen that no changes have occurred to the position or orientation of the lighting or to 10 the position of polygons from the last iteration, lighting calculations, calculated on the previous iteration, may be used again, thereby making a significant saving in processing time.

15 At step 113 the lighting of polygons, where necessary, is performed. In the system shown in Figure 2, a local transformation is performed at step 21, surfaces are lit at step 22 and a view transformation is effected at step 23. As previously stated, the local transformation and the view transformation are mathematically similar and, when performed by matrix multiplication, the matrices defining the transformations may be concatenated into a common matrix. Thus, the local transformation and the viewing transformation are 20 performed as one transformation from object space to viewing space, again significantly reducing the computational overhead.

25 Before the viewing transformation can be effected, it is necessary to perform lighting calculations. In the present embodiment, this is effected by calculating the inverse to the local transformation, effecting this inverse transformation upon the position of light sources, thereby effectively transforming the light sources into object space. Thereafter, lighting calculations are performed in object space and the object 30 is transformed directly from object space to viewing space, using the concatenated matrix. Thus, the concatenated matrix used to transform the bounding box during the object cull, at step 111, is used again to transform the actual object polygons from object space into viewing space at step 114.

35 Step 115 is substantially similar to step 24 in Figure 2 and involves the clipping of polygons which do not lie within the viewing space. However, as previously stated, additional information is generated at 40 step 111 during the object culling process which identifies planes for viewing space through which objects intersect. This information facilitates the polygon clipping process at 115, thereby again reducing compu- 45

tational overhead to effect this part of the procedure.

The projection into a 2-dimensional image at step 116 is substantially similar to the projection process performed at step 25. At step 117 back face culling is performed, as required at step 26 of Figure 2 but this procedure is performed without reference to the polygon normals, which are only used in the present embodiment within object space and are not transformed into other spaces, again reducing computational overheads. Thus, other methods must be provided to perform back face culling at step 117, as described later.

Step 118 provides scan conversion and computational overhead for this process is reduced by taking account of the restricted depth of most frame buffers, typically only having eight bits per pixel location. Thereafter, a 2-dimensional video image is produced in similar fashion to the system shown in Figure 2 by raster scanning the frame buffer at video rate.

Aspects of the system shown in Figure 11 which differs significantly from the procedure carried out in the system of Figure 2 will now be described in detail.

OBJECT CULLING

Examples of the preferred embodiments in operation will be described with reference to an example similar to that described for the conventional system.

As previously described, an object is initially defined in its own object space, as shown in Figure 3. Data stored in memory actually consists of a table of vertices defined with reference to the X, Y and Z coordinates. The object culling process at step 111 consists of examining X, Y and Z coordinates so as to identify the extent of the vertices for an object in each dimension, that is to say the maximum and minimum value in each dimension is determined. From these extents, a bounding volume in the form of a box is defined, as shown in Figure 12. Thus, plane 121 represents the maximum extent in the X direction and the plane opposite plane 121 represents the minimum extent in X. Similarly, plane 122 represents the maximum extent in Y and plane 123 represents the maximum extent in Z.

In the example previously described, the polygons in object space were instantiated three times into modelling space and the three instantiations were then transformed to viewing space. In the present embodiment, objects are transformed directly from object space to viewing space and such a transformation is carried out on the bounding box shown in Figure 12. Thus, when perspective views are being produced, the bounding boxes shown in Figure 12, are transformed into viewing space as shown in Figure 13.

In Figure 13, object 32b is completely within the viewable area. Object 32a is completely outside the viewable area and object 32c is partially inside and

partially outside.

The polygons making up object 32b must be completely processed and no saving has been made. However, the polygons making up object 32a all exist within a bounding box which is outside the viewable space, therefore, on the particular iteration under consideration, all these polygons are effectively rejected and no further processing is performed on them.

The object 32c falls within a bounding box which is partially inside the viewing area and partially outside the viewing area. As far as this object is concerned, culling must be performed on a polygon basis. The fact that this object requires polygon culling is supplied to the polygon clipping process at step 115. Furthermore, information is also supplied to step 115 stating that the object intersects the plane $X = +\text{clip}(X)$. Thus, when doing comparisons at the polygon clipping process, only clipping to the positive X direction needs to be performed, such that polygons which have X coordinates greater than $\text{clip}(X)$ are rejected while polygons having X coordinates less than or equal to $\text{clip}(X)$ are retained.

Thus, by performing the transform on the points defining the bounding box, it is possible to determine the extent to which said points are transformed into the viewing space and thereby reject, totally, any objects which fall outside the viewing space. Furthermore, it is possible, to facilitate subsequent polygon clipping of objects which are partially within the viewing space.

As noted above, although the embodiment described employs a cuboidal bounding box, any other bounding shape may be used. It should also be noted that it may not always be desirable to reject objects before transforming them into modelling space when they are outside the viewable portion of that space, for example because they are required to cast a shadow on other objects which are visible. The bounding volume test may still be useful to select objects for rejection before other aspects of further processing, however, for example to prevent effort being wasted to calculate lighting characteristics of the objects themselves.

TIME STAMPING TO AVOID LIGHTING CALCULATIONS

At step 112 changes are noted in light source parameters and changes are also noted in object positions. If a light source parameter and an object's orientation or position have not changed, it is not necessary to recalculate lighting characteristics for that particular object. Thus, lighting characteristics previously calculated are retained and used again.

Changes in lighting position, object position and object orientation are identified by applying a unique time stamp to each iteration of the process. Thus, if,

in response to interactive operations, an object or a light source is modified, said objects or light source is restamped with the unique reference for that particular iteration. At step 112 the time stamps for these objects and lights are examined against their previous value for the previous iteration, thereby identifying a condition to the effect that a position or orientation has changed.

On identifying this condition, lighting characteristics are recalculated and no saving is made. However, if, say, one object is moved in a space containing a plurality of objects, only the polygons making up that particular object require lighting characteristics to be recalculated and a significant saving can be made for the particular iteration.

LIGHTING POLYGONS IN OBJECT SPACE

a) Transforming Light Sources

In an interactive system, data is stored defining the position of light sources, a viewing position and an arrangement of polygons defining an object. In addition, a local transform defines how an object positioned in its own object space is transformed into viewable 3-dimensional space. This may be referred to as modelling space and the system is arranged such that the viewer is given the illusion of moving around within this modelling space.

In the present embodiment, the inverse of the local transform is calculated and this inverse transform is used to transform the position and/or orientation of light sources from modelling space and back into the object's own space. Thereafter, lighting characteristics of the non-transformed polygons, in their own object space, are determined in response to the transformed light source or light sources. Thereafter, the embodiment also transforms the object, lit in its own object space, under an operation which combines the local transform and the viewing transform, thus a significant saving is made, in that two transformations have been replaced by a common transformation.

Furthermore, as described with reference to Figure 5, it is necessary to consider the unit normal vector of each polygon when calculating lighting characteristics. In the present embodiment, the normal vector for each polygon is available within its own object space and the calculation using this unit vector is performed within object space. Thus, once lighting characteristics have been calculated, no further use of the unit normal vector is required and said vector is not transformed, thereby saving computational power.

A transformation of the unit normal vectors into modelling space usually results in said vectors no longer being of unit length. Consequently it would be necessary to re-normalize said vectors, which represents a significant computational overhead. By performing the lighting calculations in object space,

where the normal vectors are available, said normal vectors do not need to be transformed into modelling space for the purpose of calculating lighting. Thus, the overhead of performing this transformation is lost along with the even greater overhead of performing the re-normalisation process. As will be described below, the present embodiment also permits culling of back facing polygons after projection into viewing space, but without the need for projected surface normal vectors.

Figure 14 shows a 2-dimensional representation of object space axes X and Y, with similar modelling space axes overlaid. A polygon 141 is shown, which represents the position of the polygon in its own object space. Data is stored representing the position of vertices in object space, along with data representing the polygon's unit normal vector 142.

Under the operation of a local transform, transforming the polygon 141 into modelling space, a polygon 143 is generated in modelling space. It should be noted that the local transform affecting this transformation consists of a translation in the positive X and Y directions, an enlargement by factor 2 and an anti-clockwise rotation.

In modelling space, a light source 144 is present and the lighting characteristic of the polygon 143 would have been calculated with reference to the unit normal vector \underline{N} and the unit light vector \underline{L} . The unit vector \underline{N} would be obtained by transforming unit vector 142, which produces a vector having a length greater than unity. Consequently, it would have been necessary to perform a process of re-normalization, in order to calculate the true normal unit vector \underline{N} .

In the present embodiment, lighting calculations are not performed with reference to a transformed polygon 143, therefore, there is no need to transform unit normal vector 142. As previously stated, the inverse transform to the local transform is calculated and this inverse transform is used to effect a transformation of the light source 144 to position 145 within object space. Thus, the relationship between the inversely transformed light source 145 and the non-transformed polygon 141 is equivalent to the relationship between the transformed polygon 143 and the non-transformed light source 144.

The unit vector \underline{L} is calculated within object space and lighting characteristics are calculated from the dot product of \underline{N} and \underline{L} . If required, other lighting characteristics such as specular lighting could also be calculated within object space. If required, inverse transformations could also be performed on the viewing position, thereby bringing the viewing position into object space. Thus, Phong model specular lighting could be calculated within object space. However, such lighting effects are not calculated according to the Phong model in the present embodiment, as detailed later.

Thus, as previously stated, significant computa-

tional savings are made by effecting the inverse local transformation upon the light source rather than transforming the polygons into modelling space, so as to calculate lighting characteristics. The need to transform the unit normal vector into modelling space is avoided and a local transform and a viewing transform may be concatenated, so that only one operation is required to transform polygons from object space directly into viewing space.

b) Specular Highlights Listing

As described with reference to Figure 5, the calculation of lighting characteristics created by specular highlights is computationally demanding because values are usually raised to a power, so as to represent the way in which specular highlights rapidly fall off as movement is made from the position of highest intensity.

In the majority of applications, objects usually have few specular highlights, although such highlights add significantly to the quality of the final image. Some polygons may have many small specular reflections but very little impairment to the quality of the image results if such small insignificant highlights are ignored. Thus, for many objects, not all polygons have significant levels of specular reflection. In most cases, the majority of polygons will not have significant levels of specular reflection. In the present embodiment, polygons which have significant levels of specular reflection are grouped separately from those which do not, and are processed in accordance with a first algorithm, while the other polygons are processed in accordance with a second algorithm.

In the embodiment, the polygon data is arranged in a list so that polygons having a significant level of specular reflection are grouped together, such that all polygons requiring the first (expensive) algorithm are processed together and all polygons requiring the second (less expensive) algorithm are also processed together. By organising the polygons in this way, an algorithm or process for calculating specular reflections is only employed for polygons which have a significant amount of specular reflection thereon. As used herein, specular reflection relates to all types of reflection which requires a significant amount of processing of this type and thus may be separated from polygons which require only linear calculations for their lighting characteristics, predominantly diffuse lighting characteristics, to be calculated.

Particular savings arise by this means when generating a sequence of images interactively, or to record a motion picture sequence. Whereas known systems test each polygon every time it is rendered to determine whether the expensive specular reflection calculation is required, the present embodiment needs to perform this test only once for each polygon, for so long as its specular reflection characteristics re-

main unchanged. It will be appreciated that the surface reflection characteristics of objects are not normally required to change in the course of an interactive sequence.

5 Therefore, for each polygon and each iteration, the present embodiment avoids a memory access which would conventionally be required to read the specular reflection coefficient of the polygon and a comparison operation for determining whether the 10 level of specular reflection is significant. Those skilled in the art will appreciate that, when a large number of polygons are to be reproduced at an interactive rate, eliminating a memory access and a comparison operation can lead to a significant reduction in computational overhead.

15 It should be noted that there is not conflict between the grouping of polygons according to their specular reflection characteristics does not conflict with any requirement that polygons are sorted into 20 depth order for hidden surface removal, as will often be required. The polygon data itself is not physically moved in order to perform the ordering. Rather, lists of references to the individual polygons are kept, and a separate list can be kept for the purposes of writing 25 calculations and hidden surface removal.

c) Highlight Value Calculation

30 The Phong model for calculating specular reflection is detailed in Figure 5 and, as shown, it is necessary to calculate the dot product of the unit normal vector N with a unit vector H positioned half way between the vector L, directed towards the light source 41, and a vector E, directed towards the viewing position 42. Thus, implementing the Phong model is computationally expensive, given that the H vector must be calculated and the dot product of H with N must then be raised to a power.

35 In the present embodiment, lighting calculations 40 are performed within object space, as previously described, therefore calculation of the H vector involves transforming the viewing position into object space and then performing additional calculations in order to determine the position of said vector.

45 Strictly speaking, specular highlights refer to highlights which are dependent upon the viewing position. Thus, in the Phong model, as the viewing position is changed, the position of the specular highlights also changes, the nature of such highlights being such that they have a relatively high intensity when the viewing position is aligned with the reflection of the light source, while the intensity diminishes rapidly at other orientations.

50 55 Specular highlights significantly increase the quality of the final image by giving it a sense of realism. However, as a result of experimentation, it has been established that it is the non-linear property of these highlights, the fact that they taper off very rap-

idly from a high intensity peak, that is the major factor in creating the effect. Furthermore, colours tend to de-saturate towards the colour of the light source. Thus, an observer is often aware of the highlights as a feature of the object itself and not as a feature of the relationship between the viewing position and the light source.

The present embodiment makes use of this observation, in that lighting characteristics are calculated in response to lighting parameters of surfaces and the parameters of light sources, without making reference to the viewing position. Thus, a lighting parameter is non-linearly processed to simulate highlight characteristics, without reference to the view position.

In the embodiment, the \underline{H} vector is not calculated and a specular highlight is simulated by considering the dot product of \underline{N} with \underline{L} , as required for calculating diffuse characteristics.

To simulate the non-linear nature of such highlights, the specular characteristic is calculated by raising the dot product of \underline{N} with \underline{L} to a power n , in which n represents the specularity or shininess, of the polygon under consideration.

After calculating the non-linear term, said term is multiplied by the specular coefficient $K(s)$ and a factor representing the intensity of the light source $I(l)$.

As previously stated, in the embodiment, the specular calculation is only performed for polygons having a significant specular component. Thus, when $K(s)$ falls below a particular threshold it is not necessary to perform the specular calculations.

d) Calculating Accuracy

Referring to Figure 4, illustrating objects 32a, 32b and 32c, light source 41 and a viewing position 42 in modelling space, the positions of objects within this space are calculated to a high level of accuracy, using floating point arithmetic. A high level of accuracy is required because, given that many manipulations are performed, quantising errors would build up and could produce noticeable effects in the output image. Furthermore, the use of floating point arithmetic allows objects to be scaled over very large ranges, such that a polygon forming part of an image may be viewed at one extreme (taking up substantially the whole of the viewing field) and at another extreme, being hardly visible within a very small object.

In conventional systems, as previously described, all calculations are performed using floating point arithmetic of this type, including lighting calculations. However, although a very high level of accuracy may be provided for calculating lighting, most of this accuracy is truncated as data is supplied to the frame buffer 19, given that said frame buffer will have a limited depth, typically of eight bits per pixel location. Other systems are known in which all calcula-

tions are performed in fixed point arithmetic, but with a very large number of bits for high accuracy. These systems are generally inconvenient because of the finite world size and restricted scope for scaling etc.

5 In the present embodiment, as previously stated, means are provided for calculating the position of objects in space in high accuracy floating point arithmetic. However, lighting characteristics are calculated using lower accuracy fixed point arithmetic. The limitations of fixed point arithmetic are tolerated because results are still produced which are more accurate than the accuracy of values supplied to the frame buffer, therefore truncation is still required. However, the rate at which fixed point arithmetic may be performed is substantially higher than the rate of performing floating point arithmetic, thereby reducing computational demands.

10 In the embodiment, the fixed-point arithmetic is performed using 32 bits, of which 16 bits are allocated to whole numbers and 16 bits are allocated to fractions. The type of arithmetic employed is often referred to as integer arithmetic although, as will be appreciated, values perceived as integers by the arithmetic unit may actually represent fractions as previously described.

15 In the present embodiment, all matrix and position calculations are carried out in floating point arithmetic. Thus the position of a polygon's vertices, the location of the light sources within a world and the matrix containing the local transform are calculated in floating point arithmetic. The translation from floating point to fixed point takes place at only two places, as will now be described.

20 In a pre-calculation phase, the first point at which conversion from floating point to fixed point is required is when the normal to the polygon (or vertex) is pre-calculated. Here the algorithm is straightforward: carry out all calculations in floating points until the final values are ready for storage, then (and only then) convert the values to fixed point numbers. In some embodiments this will impose a restriction upon the coordinate system of the 3d world thus handled, in that no coordinate can exceed the largest number representable in the chosen fixed point numeric range.

25 Secondly, when lighting must be recalculated, the inverse transform of the light vector (lighting direction), with respect to the local transform of the object, is calculated in floating point to produce a floating point 3-vector. For a point or conical light source (that is, one not positioned at infinity) the inverse transformed position of the light is also calculated (also using floating point arithmetic). Then the inverse transformed direction vector is converted from floating point to fixed point. Likewise the position, for point and conical lights.

30 The normal to the polygon (in the case of facet shading) or the normal to the vertex (in the case of

vertex shading) has been precalculated in fixed point, and this is combined with the converted lighting vector in the traditional way to produce a fixed point coefficient of illumination. In the case of facet shading the coefficient of illumination is converted into a colour value (palette index or RGB triple) and stored as the colour of the polygon. In the case of vertex shading the illumination coefficient for the vertex is stored, as an integer. Interpolation to obtain per-pixel colour values can then be performed in fixed point or integer arithmetic, to achieve a significant saving. In both cases the only final stage conversion required is that of fixed point number to integer, a relatively trivial operation.

CULLING 2-DIMENSIONAL BACK FACES

In the conventional system, as previously described, back face culling is necessary once an image has been projected into 2-dimensions, that is to say, data representing points in 3-dimensions are converted to data representing the points in 2-dimensions.

In 2-dimensions, a plurality of polygons may be present at the same location. However, polygons which represent a back face are not actually visible and therefore said polygons must be culled from further processing.

In the conventional system, data representing the unit normal vector for each polygon also undergoes transformations. As previously stated, a transformation from object space into modelling space of unit normal vectors is necessary in the conventional system because this vector is used for calculating lighting characteristics. However, in the present embodiment, given that lights are back transformed into object space, it is not necessary to transform the unit normal vectors for the purpose of lighting. Therefore these vectors are not already available as data representing points in viewing space, nor are they available prior to projecting polygons into 2-dimensions.

In the present embodiment, it is still necessary to determine whether 2-dimensional polygons are forward facing or back facing, so as to effect culling of the back facing polygons. In the present embodiment, projected vectors defining the polygon edges are processed in viewing space so as to determine whether a polygon is forward facing or back facing.

The vectors compared must share a common vertex, therefore a common vertex is selected. As described with reference to Figure 10, the process of scan conversion involves selecting the vertex 101 which has the minimum y coordinate, that is to say, it is the vertex which is closest to the top of the viewing screen. As part of the scanning process and as previously described, it is necessary to calculate the gradient of the vector connecting points 101 and 102, and to calculate the gradient of the vector connecting point 101 to 103. In the table of data, due to anticlock-

wise ordering convention, point 103 will be listed prior to point 101, which in turn will be listed prior to point 102. Thus, for a forward facing polygon, irrespective of its transformation within 3-dimensional space, the vector connecting point 101 to point 102 should always be to the left of the vector connecting point 101 to point 103, when the polygon is front facing.

Thus, the gradient, that is to say the rate at which x changes with y, will always be greater (more positive) for the vector connecting vertex 101 to vertex 103 than the similar gradient for the vector connecting vertex 101 to vertex 102, when the polygon is front facing. Thus, if the gradient of the vector connecting the highest (minimum y value) vertex to its preceding vertex is less than or equal to the gradient connecting said highest vertex to its succeeding vertex, the polygon is to be culled. This is because it is either back facing or it is on edge, neither of which should be rendered.

As a numerical example, the coordinates (6, 2) (1, 8) and (12, 6) are shown in Figure 10 as the x, y positions of the vertices 101, 102 and 103. Starting at vertex 101, the vector connecting vertex 101 to vertex 102 represents a decrease of 5 in the value of x, for an increase of 6 in the value of y. This corresponds to a gradient of -5/6. The vector connecting vertex 101 to vertex 103, on the other hand, represents an increase of 6 in the value of x, for an increase of 4 in the value of y. This translates to a gradient of -6/4 or 3/2. This is greater than the other gradient -5/6, confirming that the polygon is forward facing.

It may be noted that the gradients thus calculated are not wasted, in that they are useful for the operation of scan conversion, to be described below. Moreover, where a vector forms an edge between two polygons to be considered, the calculated gradient can be stored and used for both polygons. Note also that the trivial case of Figure 10 can be confirmed as a front facing polygon without calculating the gradients, by observing that from vertex 101 to vertex 103 the value of x decreases, while from vertex 101 to vertex 102 the value of x increases. This confirms that one gradient is less than zero while the other is greater than zero, such that the one must be greater than the other. Only when both gradients are of the same sign is it necessary to perform the division and calculate the gradient. On the other hand, in an embodiment where the gradients are anyhow required for scan conversion, this pre-test on the basis of signs alone will not save any computation.

As an alternative to comparing gradients, in an alternative embodiment, z component of the vector or "cross" product of vectors connecting points 103 to 101 and 101 to 102 is calculated. Such an approach is similar to transforming the normal vectors, as previously performed. However, the magnitude of the vector is irrelevant and only its direction in the z dimension is required, thereby making the calculation

less intense than that required for re-normalization.

Thus, a partial cross product is calculated to identify the z component of a normal vector in screen space and the polygon is culled if the vector calculated in this way points into the screen. Using again the example of Figure 10, the first vector has components (-6, -4) and the second vector has components (-5, 6), in accordance with the ordering of the vertices 103, 101, 102. Representing these as (x₁, y₁) and (x₂, y₂) respectively, the z component of the cross product is defined as x₁y₂ - x₂y₁. In the present example this results in -36 - 20, which equals -56. Since this is a negative z value, pointing out of the screen, the polygon is front facing.

Note again that in many cases the multiplication of the vector components becomes unnecessary, if initial tests are performed on the signs of the four components. For example, if x₁ and y₂ are both positive or both negative, and if x₂ and y₁ are of opposite sign, the resulting z component will always be a positive value. Only in cases where the two terms x₁y₂ and x₂y₁ are of the same sign is it necessary actually to perform the multiplication and subtraction operation to determine the sign of the cross product.

It may also be noted that, compared with problems often encountered in known systems, the back facing test based on the projected vectors employed in the present embodiment automatically takes into account perspective effects included in the viewing transformation. Referring for example to the object 32b in Figures 8 and 9, the side faces of the object are clearly identified as back facing in the viewing space, whereas in the modelling space without perspective correction (Figure 6) these faces are parallel to the viewing direction.

SCAN CONVERSION

Referring to Figure 11, at step 116 the 3-dimensional view is projected to define an image in 2-dimensions, wherein each vertex is given a 2-dimensional coordinate position. At step 117 back face culling is performed as previously described so that, at any position on the screen, only one polygon is present, thereby uniquely defining the characteristics of that position.

Step 118 consists of scan conversion, which involves specifying which pixels are required to be modified to define a particular polygon and, additionally, to specify a colour for those particular pixels.

In the preferred embodiments, the process for identifying which particular pixels are to be included within a polygon is substantially similar to the process employed in the conventional system and as described with reference to Figure 10. The process is also described in the applicants co-pending European Patent Application EP-A-0531157 (not published at the present priority date) included herein by reference as

part of the present disclosure.

As previously described, it is necessary to determine gradients, specifying how x varies, that is the span across each line, as y varies on a scan line by scan line basis. Thus, gradients, conventionally referred to as variations of y with respect to x are, in accordance with the present convention, referring to variations in x with respect to y. Thus, as previously stated, as y is incremented, identifying descending scan lines, it is possible to identify pixel locations within the polygon which require colour modification.

In the present embodiments, each pixel location within the frame buffer 19 stores eight bits and, as previously stated, in the present embodiments, lighting characteristics are calculated using fixed point arithmetic, although these fixed point values will still require further truncation before being written to the frame buffer 19. In the embodiments, 2-dimensional image data is produced, wherein pixels in 2-dimensional areas are coloured in response to calculated lighting characteristics, representing the colour of said areas. Pixel colours are stored in a look-up table, which is addressed by values read from the frame buffer.

25 In a first embodiment, values stored in the frame buffer have a predetermined number of bits representing colour hue and a predetermined number of bits representing another characteristic of the selected colour.

30 In this first type of embodiment, the eight bit addresses to the look-up table, sequentially read from the frame buffer, are arranged such that three bits identify hue and the remaining five bits represent another characteristic of the selected hue. Thus, with three bits available, a total of eight hues may be provided, selected from the gamut of colours which may be realised by the VDU 18. Thus, for example, hues such as blue, red, green, magenta, cyan and yellow may be selected and the system may be considered as having a separate five bit look-up table for each of said selectable hues.

45 Various colour ramps are possible, according to the wishes of the system designer or user. In one example, within each of said five bit look-up tables, the other characteristic which varies is luminance, which varies linearly. This arrangement, previously known in principle, is shown graphically in Figure 16. Three bits of a value read from the look-up table represent colour, therefore one of the colours C1 to C8 is selected.

50 The remaining five bits define one of 32 luminance levels. Thus, linear interpolation between colours is not possible and colour aliasing occurs. However, luminance is controllable over a 32 bit range, which produces results of acceptable quality, given that the eye is more sensitive to luminance aliasing.

55 A colour ramp for an alternative example is shown in Figure 17. It should be noted that this colour ramp

is repeated eight times, in a similar fashion to Figure 16, once for each selectable colour.

In Figure 17, luminance is varied in response to the additional five bits but the variation of luminance with values read from the frame buffer 19 is non-linear to provide gamma correction, thereby allowing look-up tables for colour conversion and gamma correction to be combined into a common look-up table. This too is previously known in principle.

Another alternative colour ramp is shown in Figure 18 which again would be repeated eight times as shown in Figure 16. In Figure 18, the other characteristic of a selected colour is saturation. Thus the colour ramp may initiate at a fully saturated colour value, with a predetermined amount of luminance and, as input values increase, the saturation decreases, so that the output colour fades from a distinct colour towards white.

The purpose of the colour mapping shown in Figure 18 is to simulate the effect of specular highlights which appear substantially white (assuming white light sources), even on coloured objects. Again, this response is made non-linear which may facilitate the calculation of specular highlight non-linearities. Thus, as previously described, specular highlights may be calculated by raising the dot product of N and L to a power. In the present alternative embodiment, it may only be necessary to calculate the dot product without raising it to a power, because the non-linear raising to a power may be performed by the look-up table, when pixel values stored in the frame buffer are converted to real colour signals.

Alternatively, a specular highlight process may perform easily implemented power raising (to the power 16 say) with further non-linearity being introduced by the look-up table.

A yet further alternative embodiment for the colour ramps is shown in Figure 19, which combines the luminance characteristic of Figure 17 with the saturation characteristic of Figure 18. As previously described, three bits select one of the eight available hues. The remaining five bits represent the level of lighting applied to the surface. Initially, luminance values start at zero and the object is represented as black. As the lighting value increases, the luminance value increases and saturation remains high, such that a red area, say, gets brighter but remains red, representing the effect of diffuse reflection, until at about half way, or preferably further, luminance reaches its maximum value.

As the lighting level increases, it is assumed that this increase is due to specular reflection. Thus high input values result in a decrease in saturation, so that a red highlight changes to white. Thus, each ramp undergoes a change from black to full colour to white, wherein a different colour is provided for each ramp.

As shown, the luminance region varies non-linearly to take account of gamma correction and the sat-

uration region varies non-linearly to represent specular reflection. Alternatively, either or both of these ramps could be linear.

Figure 20 illustrates separately the green (G) blue (B) and red (R) components stored in the colour look-up table for one of the colour ramps. In particular the colour ramp illustrated in Figure 20 corresponds to the colour cyan, which is chiefly composed of green and blue components, with little or no red. As shown along the horizontal axis in the three graphs, an increase in the look-up table index over the range of the cyan colour ramp causes a change from a black colour at K, through increasingly luminous cyan shades to a maximum luminance of cyan shade at C, followed by a decrease in saturation accompanied by a continuing increase in perceived brightness, reaching bright white at point W.

This is achieved in that, from point K to point C, both the green and blue components rise strongly to a high, but not maximum, value, while the red component rises only slightly. Thus at C, large and equal proportions of green and blue are mixed with a small proportion of red, to achieve the desired high intensity, saturated cyan colour. Thereafter in the portion of the range from C to W, the red component rises exponentially, while the green and blue components also rise noticeably to a maximum value. Thus the colour desaturates towards white, and the perceived brightness of the colour continues to increase.

In many systems, the full range of colour look-up table values will not be available, for example because the operating system requires certain colour values to be permanently assigned. In an embodiment operating under Microsoft Windows (TM), for example, it will often be the case that the first ten or so values in the colour look-up table are permanently assigned for the display of borders, menus, and other messages. In such cases, it may be preferred to divide the available range of index values into sub-ranges which are not exactly powers of 2 in size. For example, 8 sub-ranges of 28 values each can be provided occupying 224 entries in the look-up table, leaving 22 entries for system purposes. In such an embodiment, rather than having separate bit fields for the colour hue selection and the brightness level index, and simply concatenating these to generate the final table index, it will be preferable to store a base index for each hue, and add to this a level value in the appropriate range in order to address individual shades within a sub-range of the look-up table. There is little loss of performance associated with this change, in that most processors can perform an add operation at the same speed as an OR operation.

For example, assume that lighting calculations have produced an illumination coefficient in the range 0.0 to 1.0 for a polygon, which is to be converted to an integer colour look-up table index, to reproduce the colour of the polygon at that particular illumina-

tion. It is a simple matter to store with the polygon the base index of the appropriate colour ramp, so as to define the basic colour (hue) of that polygon. Also recorded in the system is the size of each colour ramp (28 in the example above). The calculation of the look-up table index from the illumination coefficient is as follows.

First the illumination coefficient is scaled appropriately, multiplying it by the ramp size. Then a check is made to ensure that the illumination does not exceed the maximum representable. Finally the desired index value is simply calculated by adding the polygon colour base index to the scaled illumination value.

Figure 21 illustrates schematically the operations involved to derive the colour index as described above. Firstly, lighting characteristics are calculated in the form of illumination coefficients ranging from 0.0 to 1.0, including for example an ambient illumination coefficient $I(a)$, a diffuse illumination coefficient $I(d)$ and a specular illumination coefficient $I(s)$. These are combined at 210 into a single illumination coefficient $ITOT$. At 212 this is scaled in accordance with the ramp size, and also subjected to a check to ensure that it does not exceed the maximum within the ramp size. In other words, for a ramp size of 28, values which exceed 27 after scaling are set to 27. At 214, a base index determining the hue of polygon is received from the original polygon data and added to the scaled illumination coefficient $ITOT'$. This gives the appropriate 8-bit index value $INDEX$ which is stored in the frame store at locations corresponding to the screen addresses x, y of the pixels falling within the relevant polygon.

To display the image, the index values are read from the frame store in a raster sequence and passed to the colour look-up table 218. Here they are finally converted to 24 bit colour values (for example 8 bits each r, g and b) and then to video signals, for display on the monitor. Equally, the values may be recorded as still pictures or as video images.

As a numerical example, assume that a cyan polygon is to be represented, and is illuminated with an illumination coefficient $ITOT$ of 0.3. Further, assume that the ramp size is 28, and that the 28 entries forming the cyan colour ramp begin with entry number 150 in the colour look-up table (base index = 150). The first step is to scale the illumination coefficient, that is to multiply 0.3 by the ramp size 28. This sets the scaled illumination level $ITOT'$ to 8.4. Since the cyan colour ramp occupies entries from number 150 to number 150 plus 27, a check is made to ensure that the scaling has not increased the illumination value beyond 27, in which case $ITOT'$ would be reduced back to 27. Finally the scaled illumination level 8.4 is added to the base index 150 for the cyan colour ramp, and rounded using integer arithmetic to give a colour look-up table index of 158.

By setting the coefficients of diffuse reflection to relatively low values, and exploiting the power-of-n non-linearity in the calculations of specular illumination, it is relatively easy to ensure that diffuse reflection effects do not cause the illumination values to reach the de-saturated part of the colour ramp, even when several diffuse illumination contributions (from several light sources) are combined into a single value for generating the look-up table index. Moreover, since the entire range of levels provides a steady increase in perceived brightness, the appearance will not be degraded seriously even when diffuse illumination leads to level values in the de-saturated part of the ramp.

It should also be noted, that the index value does not need to be calculated per pixel, in that it may be constant across a polygon, or may be derivable by interpolation across the polygon. In the latter case, it may be desirable to perform truncation to integer values after interpolation, however. Note also that the second type of embodiment described permits dynamic allocation of colour look-up table values and dynamic setting of ramp sizes, in a manner which is not possible when index values are rigidly divided into colour selection bits and level bits.

Many features of the preferred embodiments reduce the computational demands for a 3-dimensional interactive graphics engine. When taken together, they allow interactive 3-dimensional graphics to be generated using a common, programmable central processing unit, with effectively no requirement for purpose-built hardware. However, it should also be understood that many sub-combinations may be selected, so as to reduce the amount of purpose-built hardware required, thereby providing significant benefit.

The techniques disclosed herein are particularly suitable to interactive graphics environments. However, the techniques may also be employed in other graphics environments, such as non-interactive environments or real-time environments, where the speed of interactivity is perceived to be instantaneous.

45 Claims

1. Apparatus for generating 2-dimensional image data from data defining objects in a multi-dimensional space, the method including the step of projecting 2-dimensional polygons into a 2-dimensional screen space, each projected polygon being forward facing or back facing, characterised by:
 - 50 means for processing projected vectors defining polygon edges in screen space to determine whether a polygon is forward facing or back facing.

2. Apparatus according to claim 1, wherein said vectors share a common vertex.

3. Apparatus according to claim 1 or 2, wherein the gradients of said vectors are compared.

4. Apparatus according to claim 1 or 2, wherein at least a partial cross product of said vectors is calculated.

5. Apparatus according to any preceding claim, including means for calculating lighting characteristics in response to lighting parameters of said polygons and the parameters of light sources, characterised by:

means for non-linearly processing lighting parameters to simulate specular highlight characteristics, without reference to a view position.

6. Apparatus according to any preceding claim, wherein:

lighting characteristics of said polygons are calculated in response to data representing light sources and lighting parameters defining coefficients of specular reflection and diffuse reflection,

not all polygons have significant levels of specular reflection, and

said apparatus includes means for testing the polygons to group those polygons having significant levels of specular reflection separately from those which do not, and processing the groups in accordance with a first process and a second process respectively, such that the lighting characteristics for successive images of a sequence can be calculated by the appropriate process without re-testing each polygon as to its specularity.

7. Apparatus according to any preceding claim, including data storage means, wherein said storage means stores data defining:

(a) the position of a light source,

(b) a viewing position,

(c) an arrangement of polygons defining an object in a local space, and

(d) a local transform for transforming said object into viewable space;

said processing means further including:

means for calculating the inverse of said local transform,

means for transforming the position and/or orientation of said light source in accordance with said inverse transform, and

means for determining lighting characteristics of the non-transformed polygons of the object in response to the transformed light source.

8. Apparatus according to claim 7, wherein said processing means yet further includes means for transforming said object in a combined operation, in accordance with the local transform and a viewing transform, said viewing transform defining the viewing position.

9. Apparatus according to any preceding claim, including storage means for storing data defining:

(a) the position of a light source,

(b) a viewing position,

(c) an arrangement of polygons defining an object and

(d) a local transform for transforming said object into viewable 3-dimensional space,

wherein said processing means includes:

means for defining a bounding volume for an object in each dimension,

means for performing said local transform on said bounding volume,

means for determining the extent to which said bounding volume is transformed into a visible part of said viewable space, and

means for on the basis of said determination selecting the object for further processing.

10. A method of generating 2-dimensional image data from data defining objects in a multi-dimensional space, the method including the step of projecting 2-dimensional polygons into a 2-dimensional screen space, each projected polygon being forward facing or back facing, characterised by:

processing projected vectors defining polygon edges in screen space to determine whether a polygon is forward facing or back facing.

11. A method according to claim 10, wherein said vectors share a common vertex.

12. A method according to claim 10 or 11, wherein the gradients of said vectors are compared.

13. A method according to claim 10 or 11, wherein at least a partial cross product of said vectors is calculated.

14. A method according to claim 10, 11, 12 or 13, wherein lighting characteristics are calculated in response to lighting parameters of polygons and the parameters of light sources, characterised by

non-linearly processing a lighting parameter to simulate specular highlight characteristics without reference to a view position.

15. A method according to any of claims 10 to 14, further comprising the steps of:

(a) defining the position of a light source,
(b) defining a viewing position,
(c) defining an arrangement of polygons for
an object in a local space,
(d) defining a local transform for transforming
said object into a viewable space,
(e) calculating the inverse of said local trans-
form,
(f) transforming the position and/or orientation
of said light source in accordance with said in-
verse transform, and
(g) determining lighting characteristics of the
non-transformed polygons of the object in re-
sponse to the transformed light source. 5

15

16. A method according to claim 15, further comprising
transforming said object in a combined oper-
ation, in accordance with the local transform and
a viewing transform, said viewing transform de-
fining the viewing position. 20

17. An image signal or recording generated using a
method or apparatus as claimed in any preceding
claim. 25

30

35

40

45

50

55

17

FIG. 1

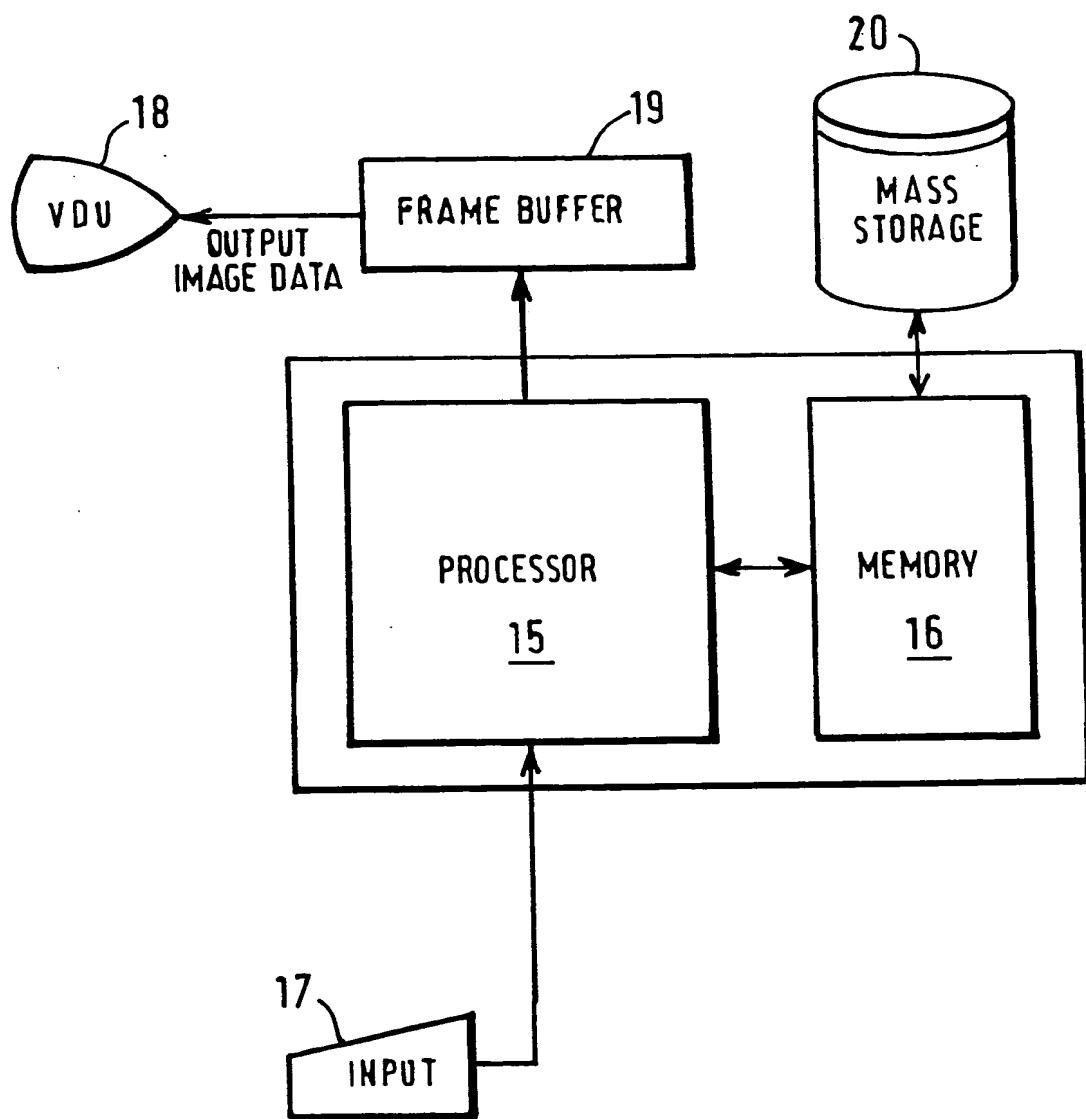
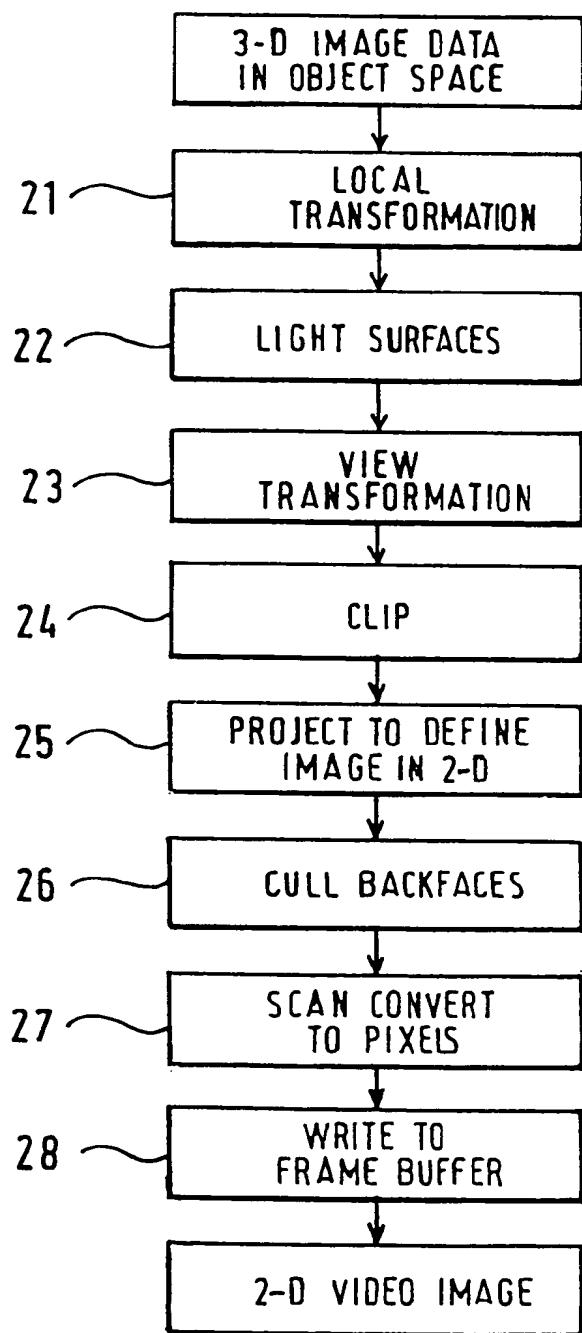


FIG. 2



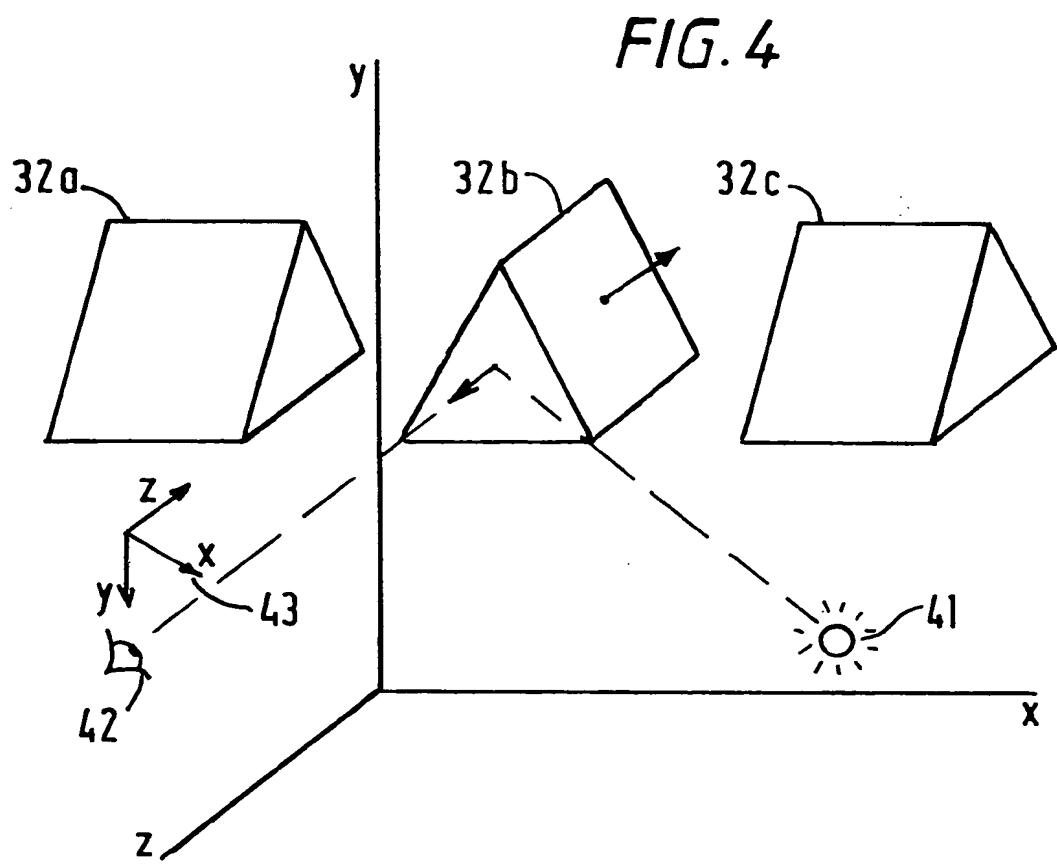
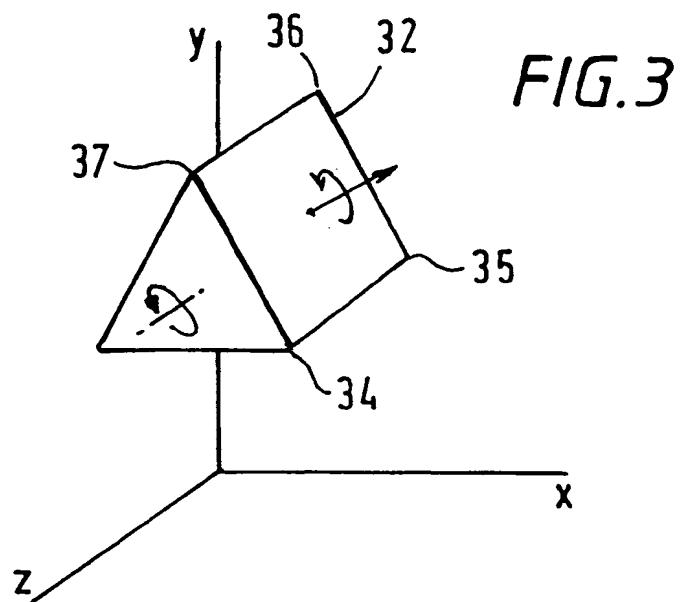
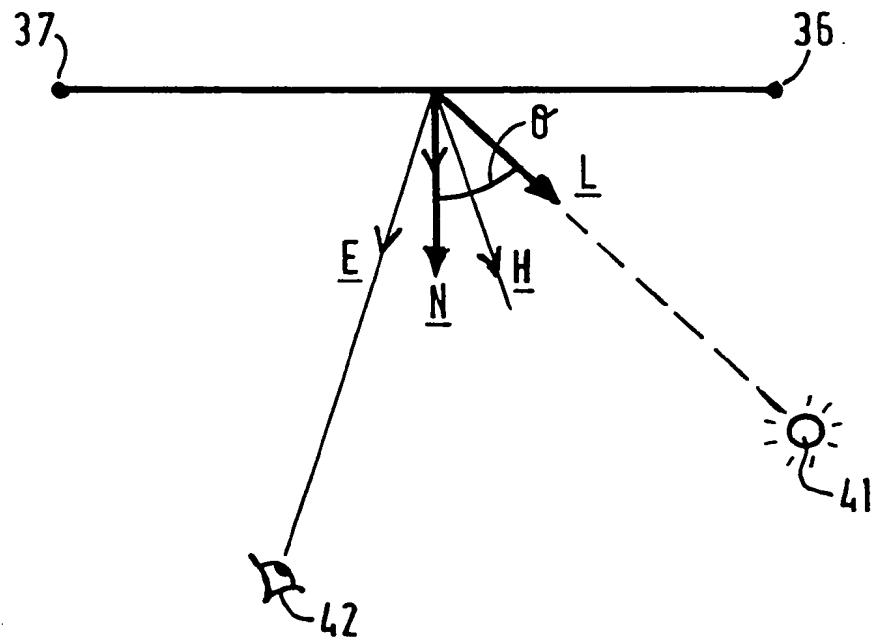


FIG. 5



$$I(a) = B(a) \cdot K(a)$$

$$I(d) = I(l) \cdot K(d) \underline{N} \cdot \underline{L}$$

$$I(s) = I(l) \cdot K(s) (\underline{H}, \underline{N})^n$$

$$I(RGB) = I(a) + I(d) + I(s)$$

FIG.6

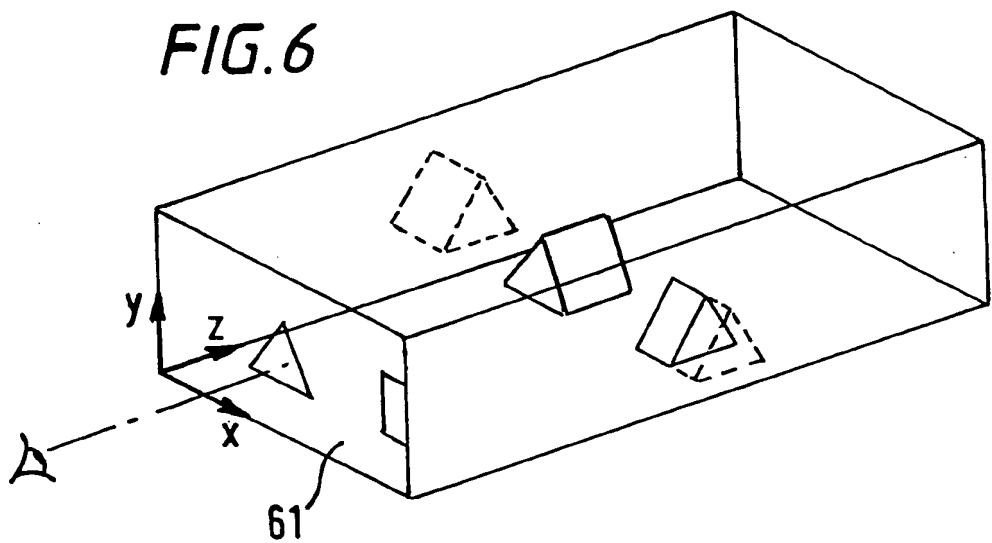


FIG.7

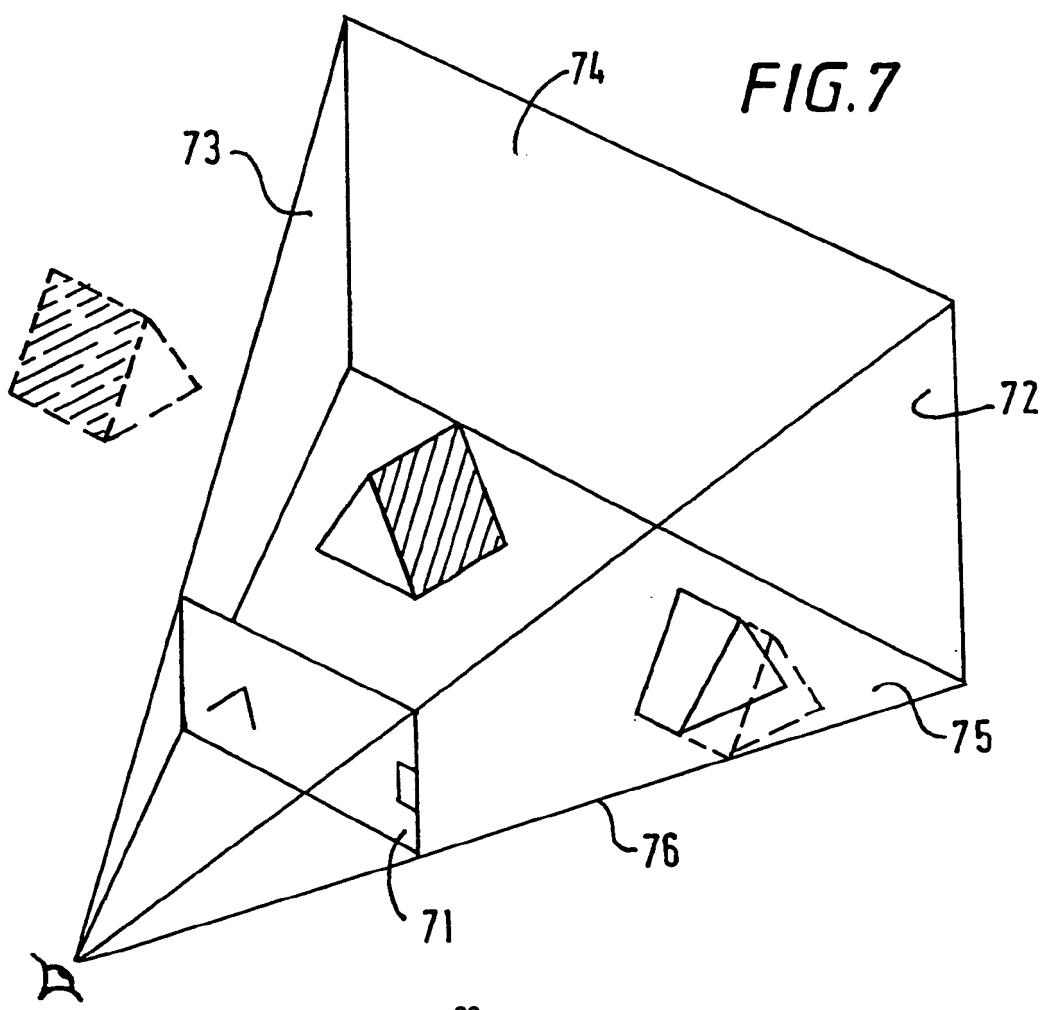


FIG.8

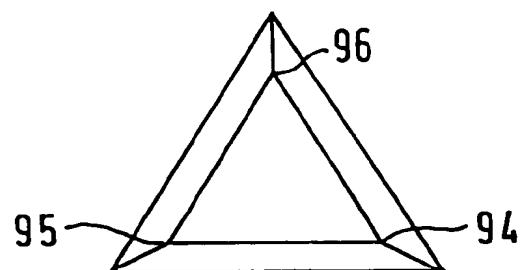
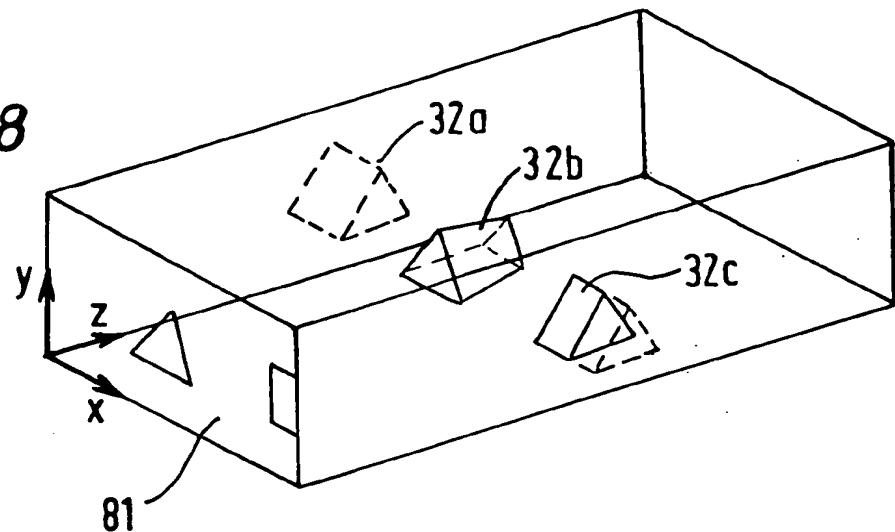


FIG.9

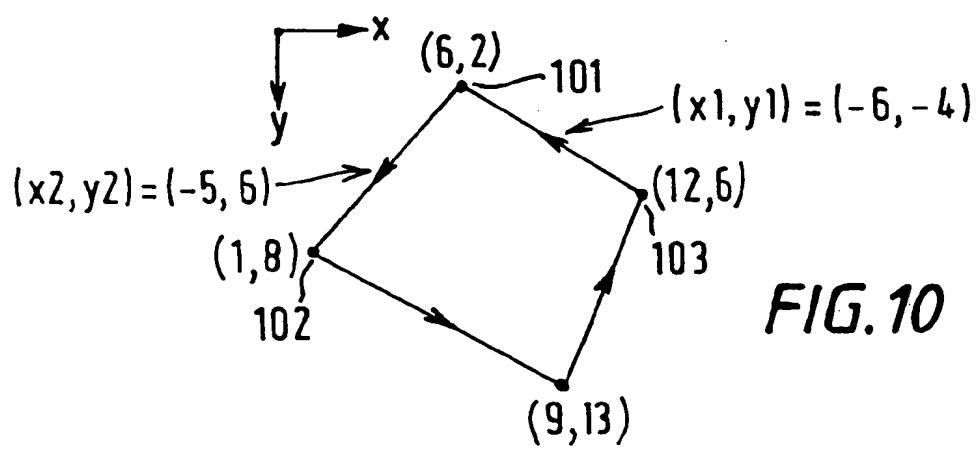


FIG. 11

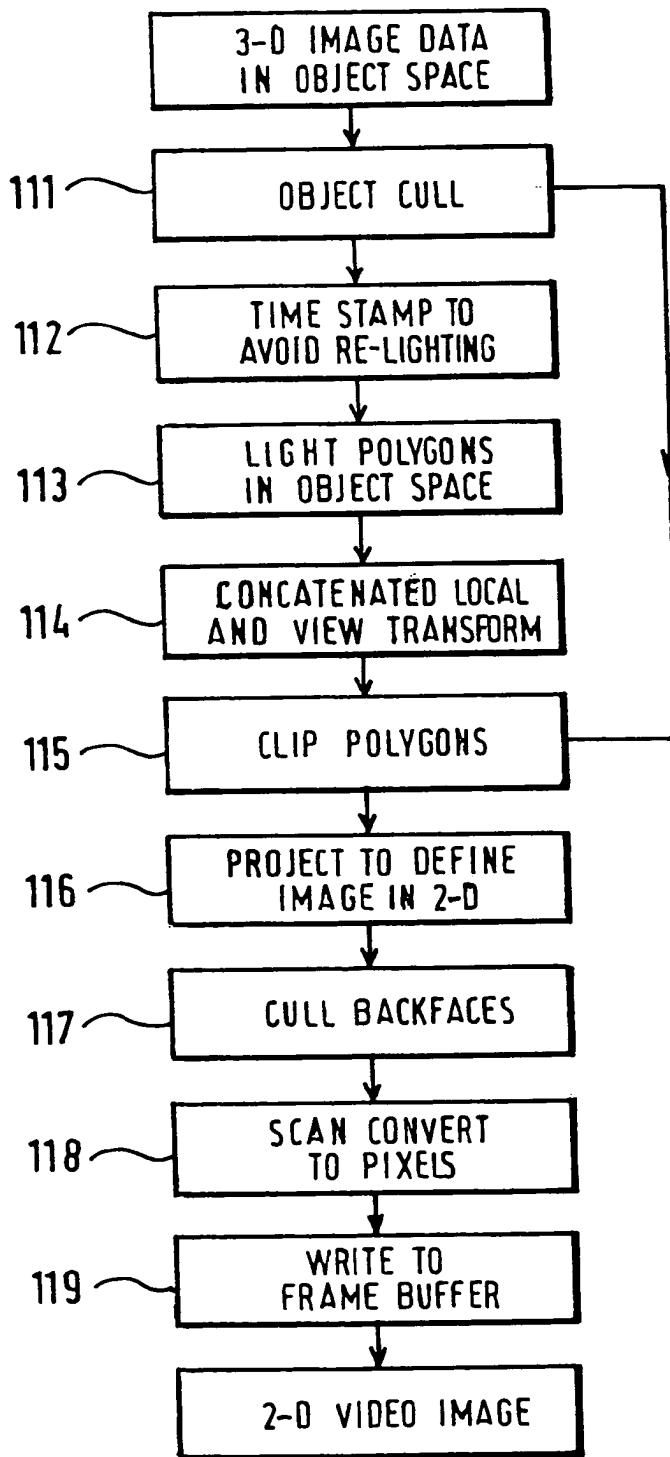


FIG. 12

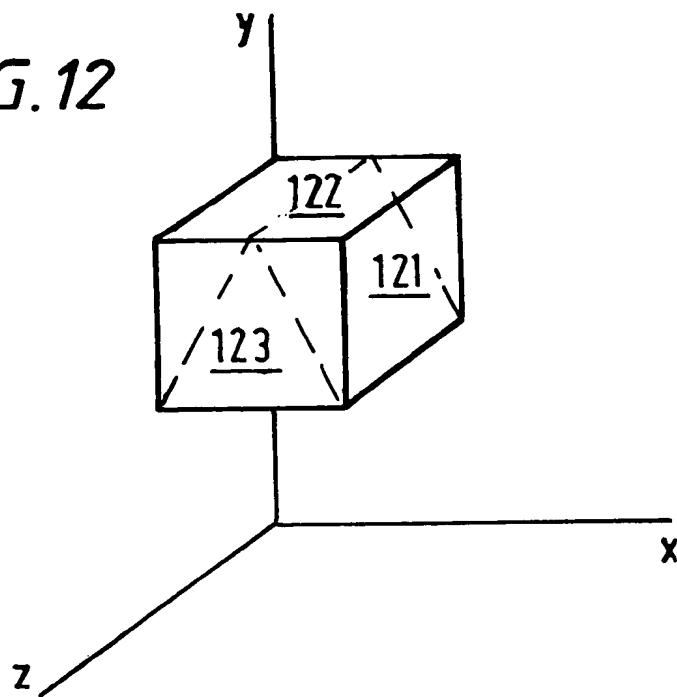


FIG. 13

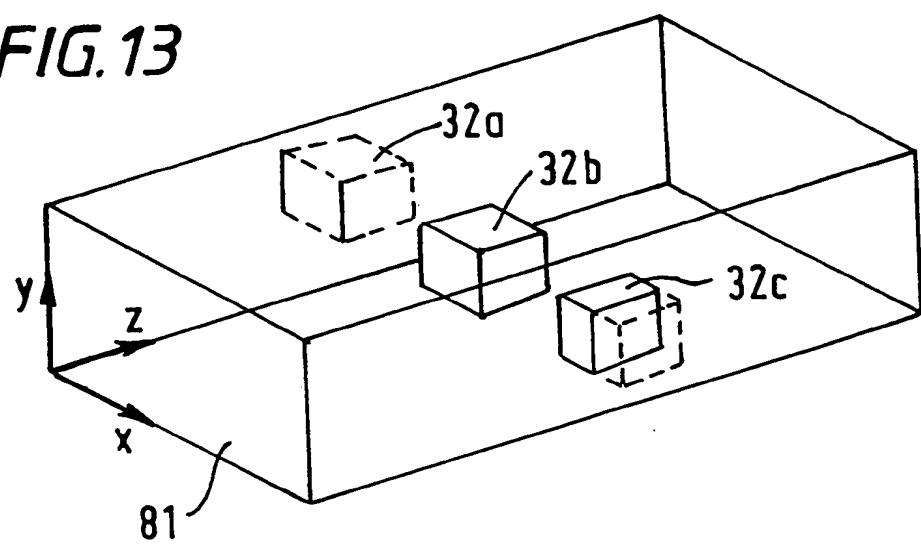


FIG. 14

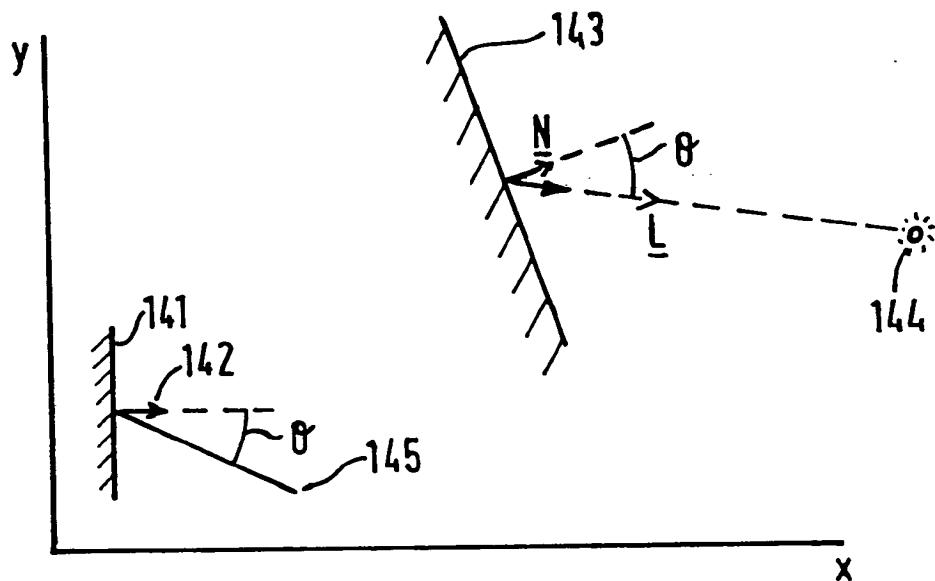


FIG. 15

$$I(d) = I(l) \cdot K(d) \underline{N} \cdot \underline{L}$$

$$I(s) = I(l) \cdot K(s) \cdot (\underline{N}, \underline{L})^n$$

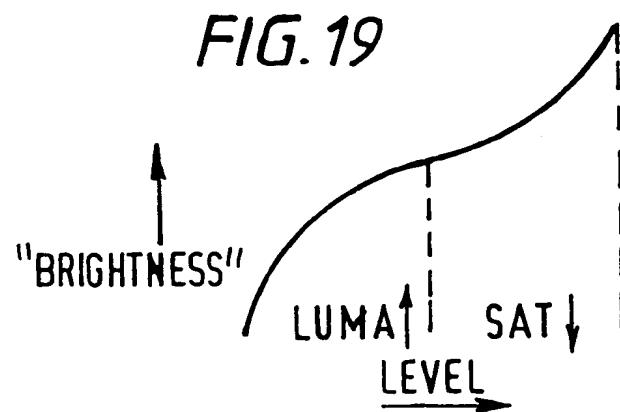
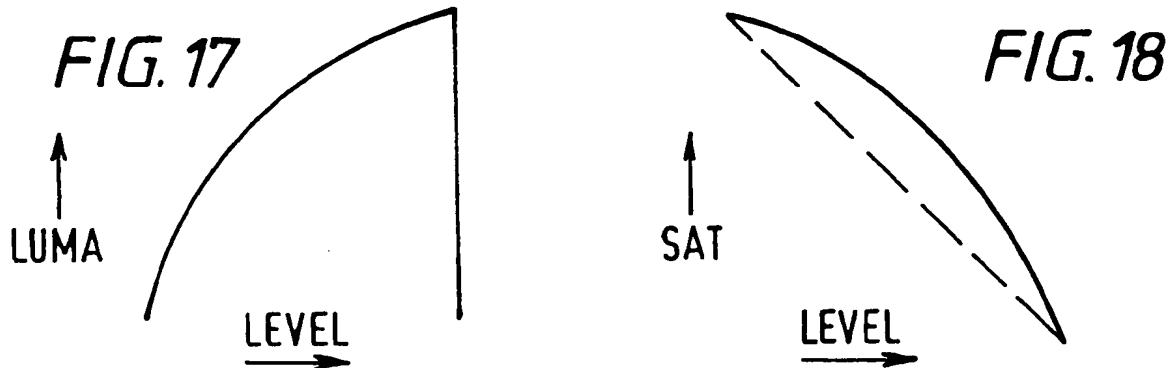
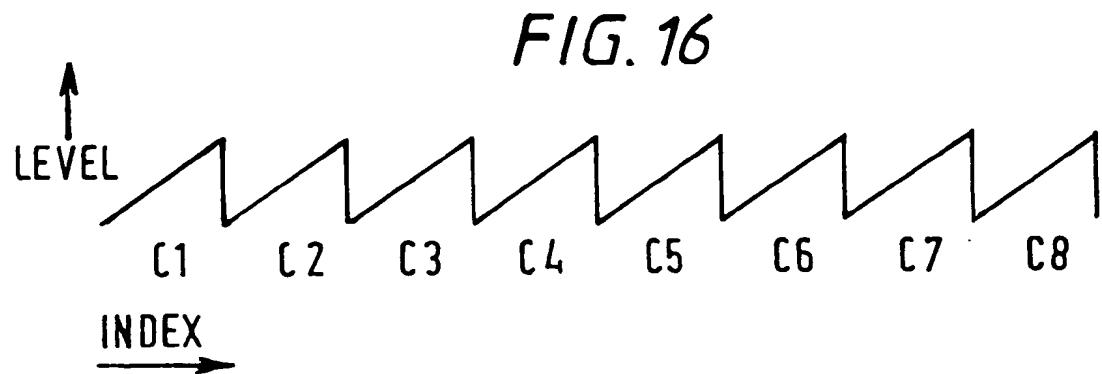


FIG.20

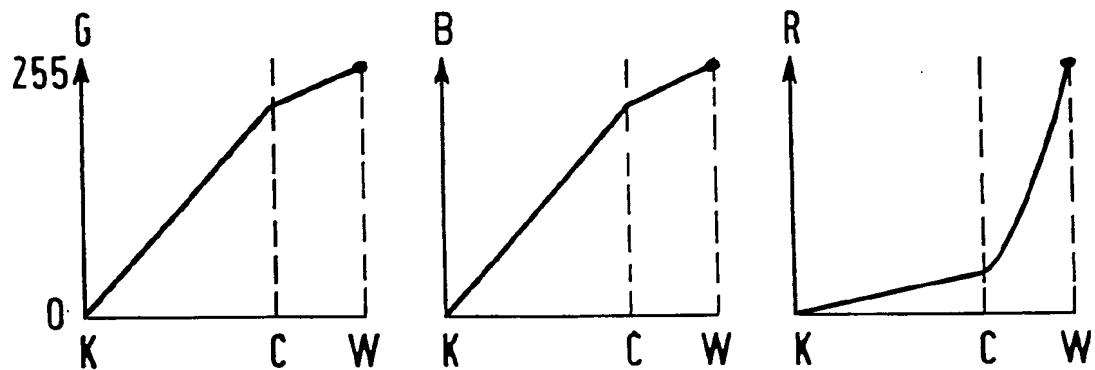


FIG.21

